In the specification:

Page 14, line 15 to line 23:

5 Customer management subsystem 130 coordinates the management of information related to the user and the users' use of voice portal 10. In an exemplary embodiment, customer management subsystem 130 acquires information on the user, such as, preferences and demographics, which are selectively used by user interface 110, advertising subsystem 120, and other functions of voice portal 10. Customer

10 management subsystem 130 can perform additional functions related to identification, session, user, and payment protocols. Although subsystem 110, 120 and 130 are described separately, the operations of each can be integrated in a single unit without departing from the principles of the invention.

15

Page 19, line 7 to line 20:

Referring still to figure 2, existent subsystem 140 coordinates access by user interface 110, advertising subsystem 120, customer management subsystem 130, fusion engine

20 150, and update engine 160 to database 170. Data contained in database 170 is gathered from Internet sources by update engine 160. In an exemplary embodiment, the data structure used in database 170 is based on a hierarchy of "existants" or things and their relationships and associations with each other. Advantageously, the ability to replicate and modify information in database 170 is more easily done because

25 database 170 interacts only with the existant subsystem 140. Existants and their creation are described further with reference to FIGURES 4-10. Specifically, an exemplary data structure model for existants is described with reference to FIGURES 4-6 although various other structures for existants is described with reference to FIGURES 4-6 although various other structures for existants can be utilized. Creation

30 and updating of existants are described with reference to FIGURES 7-10.

FIGURE 6 illustrates an exemplary data structure model 450 used by database 170 of voice portal 10 for information related to advertising. Depictions of inheritance and association relationships are the same as in the depiction of data structure model 300 in FIGURE 4. In an exemplary embodiment, advertising information arranged in data structure model 450 is contained in database 170. However, in alternative embodiments, such advertising information may be contained in a separate advertising database.

To start a new data source, a rule writer enters the data source (e.g. AMAZON Book) in a new vendor window 1120. The rule writer presses 'Enter' and clicks the 'New' button. After this action is performed, a graphical, a graphical user interface (GUI) 1300 illustrated in FIGURE 13 is shown. A URL is displayed corresponding to the selected vendor name. The rule writer is asked to confirm the correct URL. In the example of AMAZON book, a URL for a domain "AMAZONBook.com" appears. However, the domain in the URL link should read "AMAZON.com." The rule writer corrects the URL and clicks the "done" button.

Referring now to FIGURE 11, the rule writer selects the type of query that is desired. First, the rule writer selects query window 1140 and chooses from a list of potential queries. For example, "book package" may be a possible query for the book vertical domain of interest. This search is started when the rule writer clicks on the "SDE" (search data editor) button in query window 1140. The SDE button invokes a search data editor, which provides a graphical user interface (GUI) 1400 illustrated in FIGURE 14. GUI 1400 shows a list of attributes useable in a search for the particular item of

interest. For example, where books are being searched, attributes such as ISBN 1410 or UPC 1420 are shown. Where searches are for other items, attributes are listed which correspond to that item. A search for "Movie Showings" results with attributes listed, such as, Movie Package, time and showing date (see block 330 described with reference to FIGURE 14.

Page 31, line 22 to page 32, line 2:

The rule writer types a ISBN number into the corresponding data box and clicks 'Done.' Buttons 1430 in GUI 1400 advantageously allow the rule writer to save different search criteria during different searches. Once the search criteria is entered, the rule writer clicks 'Done' and because no rules have been defined for the particular data source (i.e., AMAZON Book), a graphical user interface (GUI) 1500 illustrated in FIGURE 15 appears. GUI 1500 asks whether the rule writer wants to add a new rule or change the search data. In this example, the rule writer clicks on the "add" button and GUI 1500 expands to become graphical user interface (GUI) 1600 illustrated in FIGURE 16.

Page 32, line 3 to line 9:

Referring now to FIGURE 16, the rule writer confirms that the correct type of query is highlighted. In this example, ISBN is highlighted and the rule writer clicks on the "yes" button. A graphical user interface (GUI) 1700 illustrated in FIGURE 17 appears to instruct the rule writer that the home page of AMAZON Book is loaded into the netscape browser. The rule writer is instructed to browse the web page associated to the ISBN rule. Once the search page is loaded into the Internet browser, the rule writer clicks the "done" button.

Data organizing tool 1025 (FIGURE 10) displays the resulting page in the Internet browser. If the page is correct, the rule writer clicks on "OK" on GUI 1900. A graphical

5   user interface (GUI) 2000 illustrated in FIGURE 20 appears and asks how to detect single items on the page if the search matches on multiple items. GUI 2000 is also used to indicate where to find the URL to get details about the queried item. If only a single item was found, the rule writer clicks the "defer" button because not enough information is present to build the regular expression. If multiple items are found, a

10   regular expression is entered into a data window 2010. For example, an author search may return multiple items because a single author may have written several books. In other cases, even if the query only matches one item, it may be necessary to follow an additional URL link to get the information.

15

FIGURE 25 illustrates an exemplary process of fusing information in a database. In exemplary embodiment illustrated by FIGURE 25, a flowchart 2500 depicts a simplistic

20   fusion process, or "quick fusion", executed by fusion engine 150 (FIGURE 2). In a step 2510, update engine 160 receives information from network 20 and places the information in a existant data structure in database 170 via existant subsystem 140, which accesses database 170. After step 2510 is performed, a step 2515 is performed in which fusion engine 150 gathers exact fusion attributes from an attribute definition

25   table corresponding to the existant retrieved in step 2510. After step 2515 is performed, a step 2520 is performed in which fusion engine 150 executes a "mash" of each fusion attribute from existants retrieved from database 170 into an easily comparable form. In an exemplary embodiment, a "mash" form removes spaces, prepositions, and other non-essential words. Advantageously, a "mashed format

30   provides for quick searching capabilities.

Figure 27 illustrates an exemplary process of creating a canonical data structure from two data structures. A data file 27 is identified by a unique identification number and contains a first data file 2710, a second data file 2720, and a canonical data file 2730. In an exemplary embodiment, first data file 2710 contains information relating to particular movie retrieved from the IMDB ("Internet Movie Database") website. Second data file 2720 includes movie information for a particular movie obtained from the REEL.com website. In the example illustrated by FIGURE 27, data file 2710 includes a title "Boys of Arizona," the director "Wiltz," the release year "1997," and a synopsis "great movie." Similarly data file 2720 includes a title "The Boys of Arizona," the director "Bob Wiltz," the release year "1998," and a synopsis which is blank.

5

10

15

20